D.8

Consultative
Committee for
Space Data Systems

RECOMMENDATION FOR SPACE
DATASYSTEMSTANDARDS

PACKET TELEMETRY

CCSDS 102.0-B-2

BLUE BOOK

JANUARY 1987

AUTHORITY

Issue:	Blue Book, Issue 2
Date:	January 1987
Location:	CCSDS Plenary Meeting November 1986 Frascati, Italy

This Recommendation reflects the consensus technical agreement of the following member Agencies of the Consultative Committee for Space Data Systems (CCSDS):

- Centre National D'Etudes Spatiales (CNES)/France.
- Deutsche Forschungs-u. Versuchsanstalt fuer Luft und Raumfahrt e.V (DFVLR)/ West Germany.
- European Space Agency (ESA)/Europe.
- Indian Space Research Organization (ISRO)/India.
- Instituto de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.

The following observer Agencies also concur with this Recommendation:

- British National Space Centre (BNSC)/United Kingdom
- Chinese Academy of Space Technology (CAST)/People's Republic of China
- Department of Communications, Communications Research Centre (DOC-CRC)/Canada.

This Recommendation is published and maintained by:

CCSDS Secretariat
Communications and Data Systems Division (Code-TS)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of seven member space Agencies. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed RECOMMENDATIONS and are not considered binding on any Agency.

This RECOMMENDATION is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this RECOMMENDATION is entirely voluntary. Endorsement, however, indicates the following understandings:

o Whenever an Agency establishes a CCSDS-related STANDARD, this STANDARD will be in accord with the relevant RECOMMENDATION. Establishing such a STANDARD does not preclude other provisions which an Agency may develop.

o Whenever an Agency establishes a CCSDS-related STANDARD, the Agency will provide other CCSDS member Agencies with the following information:

-- The STANDARD itself.

-- The anticipated date of initial operational capability.

-- The anticipated duration of operational service.

o Specific service arrangements shall be made via memoranda of agreement. Neither this RECOMMENDATION nor any ensuing STANDARD is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this Recommendation will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or cancelled.

FOREWORD

This document is a technical Recommendation for use in developing packetized telemetry systems and has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The Packet Telemetry concept described herein is the baseline concept for spacecraft-to-ground data communication within missions that are cross-supported between Agencies of the CCSDS.

This Recommendation establishes a common framework and provides a common basis for the data structures of spacecraft telemetry streams. It allows implementing organizations within each Agency to proceed coherently with the development of compatible derived Standards for the flight and ground systems that are within their cognizance. Derived Agency Standards may implement only a subset of the optional features allowed by the Recommendation and may incorporate features not addressed by the Recommendation.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This Recommendation is therefore subject to CCSDS document management and change control procedures which are defined in Reference [1].

DOCUMENT CONTROL

Document	Title	Date	Status/Remarks
--	Recommendation for Space Data System Standards: Packet Telemetry, Issue 1	May 1984	original Issue
CCSDS 102.0-B-2	Recommendation for Space Data System Standards: Packet Telemetry, Issue 2	January 1987	Current Issue (Supersedes Issue 1)

NOTE: Substantive technical changes from the previous issue are flagged with change bars in the margin.

CONTENTS

Sections Page

Figures

REFERENCES

[I] "Procedures Manual for the Consultative Committee for Space Data Systems", Issue 1, Consultative Committee for Space Data Systems, August 1985 or later issue.

[2] "Telemetry Channel Coding", Recommendation CCSDS 101.0-B-2, Issue 2, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[3] "Telecommand, Part 2: Data Routing Service, Architectural Specification", Recommendation CCSDS 202.0-B-1, Issue 1, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[4] "Telemetry: Concept and Rationale", CCSDS 100.0-G-1, Issue 1, Green Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

The latest issues of these documents may be obtained from the CCSDS Secretariat at the address indicated on page i.

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a common Recommendation for the implementation of spacecraft “Packet Telemetry” systems by the Agencies participating in the Consultative Committee for Space Data Systems (CCSDS).

1.2 SCOPE

Packet Telemetry is a concept which facilitates the transmission of space-acquired data from source to user in a standardized and highly automated manner. Packet Telemetry provides a mechanism for implementing common data structures and protocols which may enhance the development and operation of space mission systems.

This Recommendation addresses the following two processes:

(1) The end-to-end transport of space mission data sets from source application processes located in space to distributed user application processes located on the ground.

(2) The intermediate transfer of these data sets through space data acquisition networks, which contain spacecraft, radio links, tracking stations, ground communications circuits and mission control centers as some of their components.

This Recommendation is limited to describing the telemetry formats which are generated by the spacecraft in order to execute its role in the above processes. The jointly agreed CCSDS channel coding mechanisms required to implement space-to-ground data links of acceptable quality are defined in Reference [2]. This Recommendation therefore contains specifications for the following data structures:

(1) A **SOURCE PACKET,** which provides protocol data formatting services so that data may be exchanged between a source application process in space and its associated user application process(es) on the ground. The Source Packet format is defined in Section 3. Optional **SEGMENTATION** data structures are also described which permit very long Source Packets to be reformatted into shorter data units. The Segmentation options are defined in Section 4.

(2) A **TRANSFER FRAME,** which facilitates movement of the packetized or segmented source data through the spacecraft-to-ground communications path. As an alternative to Segmentation, the Transfer Frame also provides a mechanism to time-share the data link between sources by creating logical VIRTUAL CHANNELS. The Transfer Frame format is defined in Section 5.

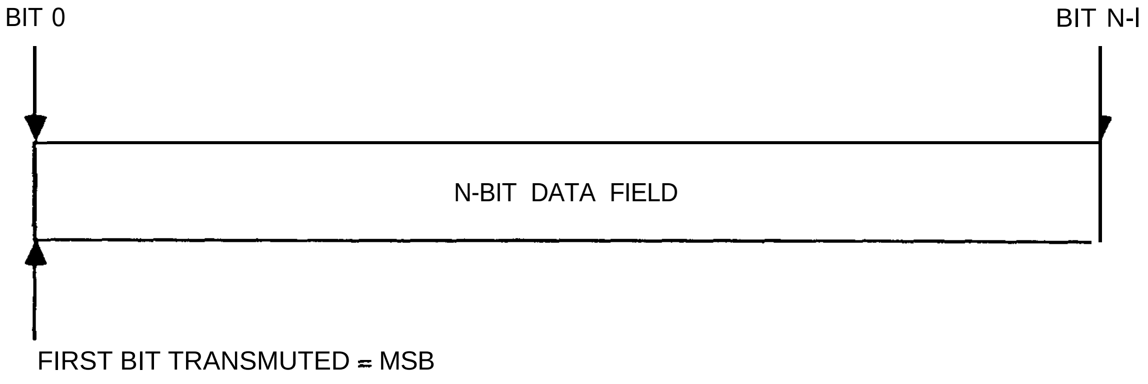
1.3 APPLICABILITY

This Recommendation applies to the future exchange of Packet Telemetry between CCSDS Agencies in cross-support situations. The Recommendation includes comprehensive specification of the structure of data streams that are generated by remote space vehicles for telemetering to space mission data processing facilities (which are usually located on Earth). The Recommendation does not attempt to define the architecture or configuration of these data processing facilities, except to describe assumed ground data handling services which affect the selection of certain onboard formatting options.

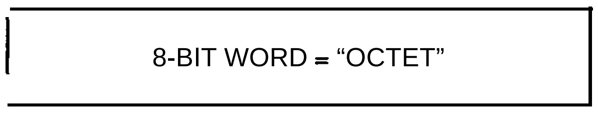
The Recommendation specifies a wide range of formatting capabilities which may facilitate a high degree of flexibility in the design of spacecraft data acquisition systems; however, compatibility with the Packet Telemetry concept may be realized by only implementing a narrow subset of these capabilities. Some "Application Notes" which discuss how different levels of compatibility may be achieved are included in the Telemetry "Green Book", Reference [4].

1.4 BIT NUMBERING CONVENTION AND NOMENCLATURE

In this document, the following convention is used to identify each bit in a forward-justified N-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be "Bit 0"; the following bit is defined to be "Bit 1" and so on up to "Bit N-1". When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., "Bit 0".



In accordance with modern data communications practice, spacecraft data fields are often grouped into 8-bit "words" which conform to the above convention. Throughout this Recommendation, the following nomenclature is used to describe this grouping:



1.5 RATIONALE

The CCSDS believes it is important to document the rationale underlying the standards chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The concept and rationale for Packet Telemetry may be found in Reference [4].

2 OVERVIEW

Figure 2-l is a functional diagram of the telemetry data flow from the creation of a data set by an application process operating within a spacecraft "source" (instrument or subsystem), through to the delivery of the same data set to a user "sink" (application process) on the ground. Since many of the elements of this flow are presently mission unique, a primary objective of Packet Telemetry is to define stable, mission-independent interface standards for the communications paths within the flow.

Data structures within a packet telemetry system are provided:

(1) To allow the user to optimize the size and structure of his application data set with a minimum of constraints imposed by the spacecraft-to-ground transport system. The user should thus be able to define his data organization independently of other users and to adapt this organization to the various modes of his experiment.

The data structure which enables this independence is the telemetry Source Packet. User data are encapsulated within a packet by prefacing them with a standard label or "primary header", which is used by the data transport system to route the data through the system. A secondary header structure is provided which enables more application-unique specification of the user data.

(2) To allow the ground data acquisition system to capture the packets in a standardized way, with a data-independent method of performing packet reassembly and determination of data quality. The data structure within the packet telemetry system for achieving this is the Transfer Frame, which provides the necessary header elements for extracting packets or segments from the frame and allows optional error-detection coding for deciding whether the frame handling process was error free.

(3) To allow the spacecraft terminus of the data transport system to be designed and tested without a detailed definition of all user data organizations. This is particularly important since at the time of specifying the onboard spacecraft data system design this knowledge is usually not available from the experimenter, and attempts to force an early agreement with present systems often lead to non-optimum fixed-format telemetry design.

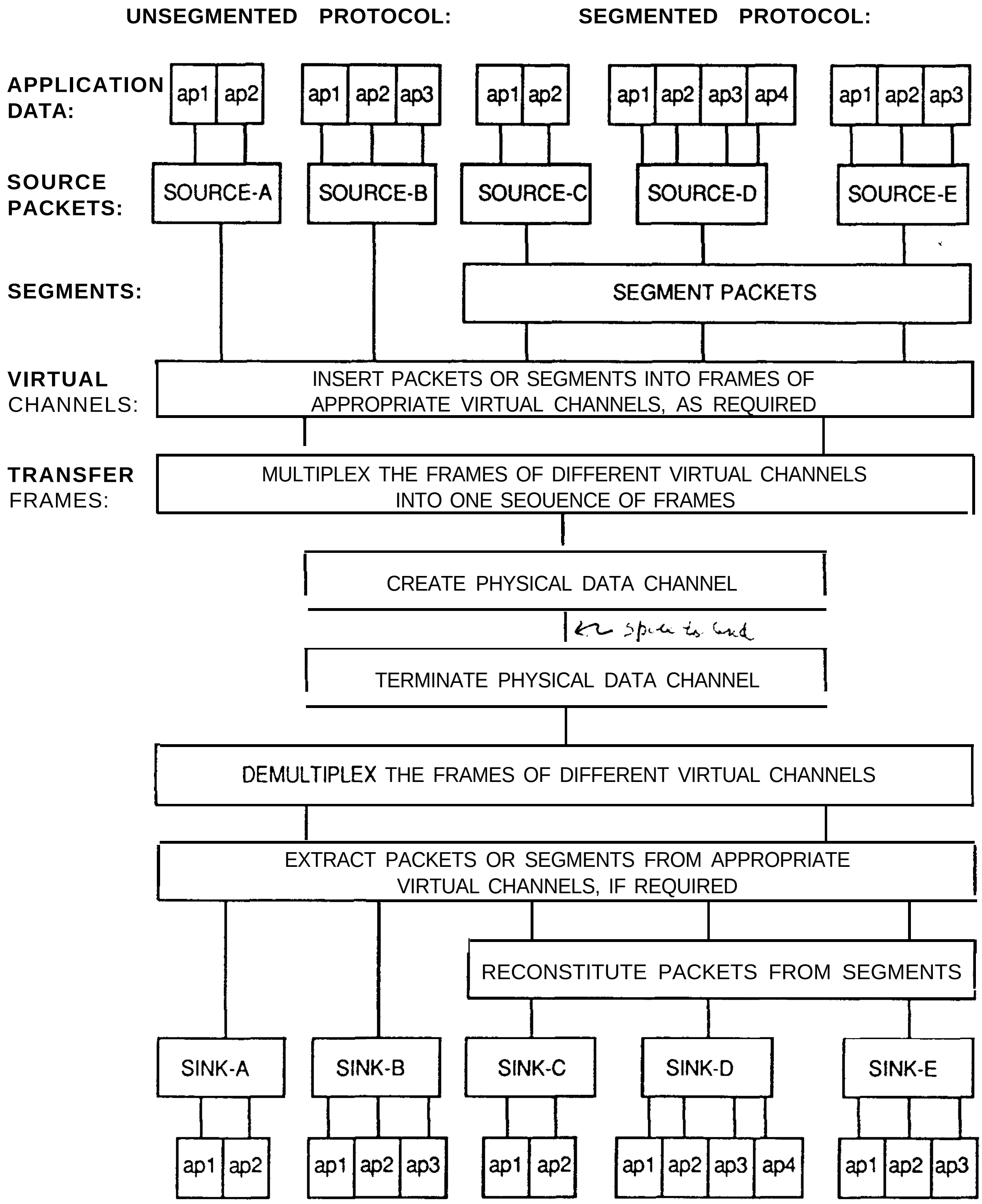

UNSEGMENTED PROTOCOL:
SEGMENTED PROTOCOL:
APPLICATION DATA:
ap1
ap2
ap3
ap4
SOURCE PACKETS:
SOURCE-A
SOURCE-B
SOURCE-C
SOURCE-D
SOURCE-E
SEGMENTS:
SEGMENT PACKETS
VIRTUAL CHANNELS:
INSERT PACKETS OR SEGMENTS INTO FRAMES OF APPROPRIATE VIRTUAL CHANNELS, AS REQUIRED
TRANSFER FRAMES:
MULTIPLEX THE FRAMES OF DIFFERENT VIRTUAL CHANNELS INTO ONE SEOUENCE OF FRAMES
CREATE PHYSICAL DATA CHANNEL
← space to gnd
TERMINATE PHYSICAL DATA CHANNEL
DEMULTIPLEX THE FRAMES OF DIFFERENT VIRTUAL CHANNELS
EXTRACT PACKETS OR SEGMENTS FROM APPROPRIATE VIRTUAL CHANNELS, IF REQUIRED
RECONSTITUTE PACKETS FROM SEGMENTS
SINK-A
SINK-B
SINK-C
SINK-D
SINK-E

Figure 2-1: Telemetry Data Flow

Since most space communications systems are capacity-limited, multiple users must be guaranteed access to the downlink data channel. It is therefore important for the spacecraft to be able to manage the data flow to the ground in an orderly manner. For example, users who generate long packets could, if allowed unrestricted access, monopolize the channel for long periods of time, forcing increased buffer capacity and response time for other users. This problem is solved by permitting two methods for controlling data flow, namely:

(a) Virtual Channelization. This is a logical mechanism that allows sources which generate very long packets to be "virtually" given exclusive access to the physical data channel by assigning them transmission capacity on a frame-by-frame basis. The data structure used to implement this feature is the **"Virtual Channel".**

Virtual channelization will normally be used to separate sources of very different characteristics. For example, if a payload contains an imaging instrument which produces a regular scan line packet containing many thousands of bits, and a number of processor-based experiments which aperiodically generate smaller packets of processed data, a possible system architecture would be to assign the imaging instrument to one virtual channel and to handle the rest by segmentation or direct multiplexing on a second virtual channel.

(b) Segmentation. Several optional mechanisms exist for transmitting long source packets as a series of shorter packets or segments, thus avoiding exclusive capture of the channel by one source. Within one option a formal data structure is introduced to implement this feature: this is the "Telemetry **Segment".**

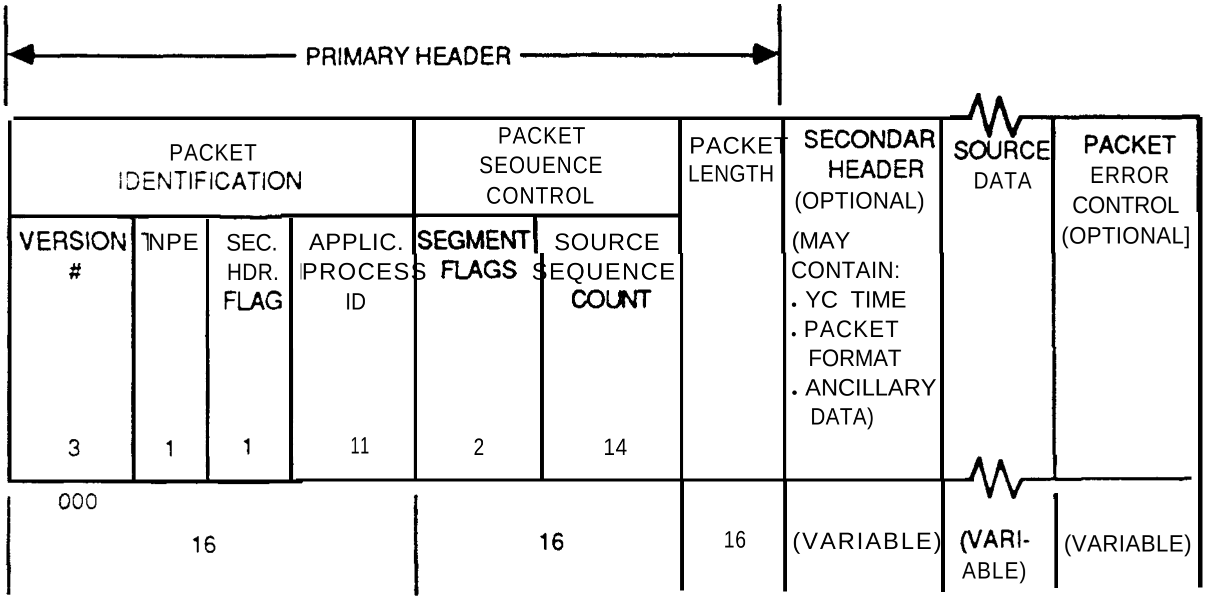
"Application Notes", which describe how compatibility with these various data structures may be achieved, are presented in Reference [4], along with key elements of the rationale behind Packet Telemetry.

3 SOURCE PACKET FORMAT

A Source Packet encapsulates a block of observational and ancillary application data which is to be transmitted from a data source in space to a source analysis facility on the ground.

The Source Packet structure permits future "versions" of the data structure to be defined, if required. The Version 1 Source Packet format, which is shown in Figure 3-1, consists of the following four major fields:

Major Field	**Length (Bits)**
Primary Header	48
Secondary Header	Variable (optional)
Source Data	Variable
Packet Error Control	Variable (optional)



PRIMARY HEADER									
PACKET IDENTIFICATION				PACKET SEOUENCE CONTROL		PACKET LENGTH	SECONDAR HEADER (OPTIONAL)	SOURCE DATA	PACKET ERROR CONTROL (OPTIONAL]
VERSION #	TNPE	SEC. HDR. FLAG	APPLIC. PROCESS ID	SEGMENT FLAGS	SOURCE SEQUENCE COUNT		(MAY CONTAIN: . YC TIME . PACKET FORMAT . ANCILLARY DATA)		
3	1	1	11	2	14				
000									
16				16		16	(VARIABLE)	(VARIABLE)	(VARIABLE)

Figure 3-1: Version 1 "Source Packet" Format

3.1 PRIMARY HEADER

The primary header consists of 48 bits subdivided into the following fields:

Field	Length (Bits)
PACKET IDENTIFICATION	16
- Version Number (3)	
- Type (1)	
- Secondary Header Flag (1)	
- Application Process ID (11)	
PACKET SEQUENCE CONTROL	16
- Segmentation Flags (2)	
- Source Sequence Count (14)	
PACKET LENGTH	16
	48

3.1 .1 PACKET IDENTIFICATION (16 Bits)

This 16-bit field is separated into four subfields:

(1) Version Number (Bits 0 through 2)

The Version Number occupies the three most significant bits of the packet Primary Header. By changing the Version Number, future variations of the Source Packet structure become possible. AT PRESENT, ONLY TWO VERSIONS OF THE PACKET ARE PERMITTED:

(a) “Version 1” (Bits O-2 = 000) is the complete Telemetry Source Packet format, which is described in the remainder of Section 3.

(b) “Version 2” (Bits O-2 = 100) is the Telemetry Segment format, which is described in Section 4.

FOR VERSION 1, THE REMAINDER OF THE SOURCE PACKET FORMAT IS DEFINED AS FOLLOWS:

(2) Type (Bit 3)

This single bit is used to identify that this is a Telemetry Packet rather than a Telecommand (TC) Packet. A TC Packet has this bit set to value "1": therefore, in all Telemetry Packets, bit 3 shall be set to "0".

(3) **Secondary Header Flag (Bit 4)**

This l-bit flag signals the presence (Bit 4 = 1) or absence (Bit 4 = 0) of a Secondary Header data structure within the Source Packet.

(4) Application Process ID (sits 5 through 15)

This 1 l-bit field uniquely identifies the individual application process within a particular space vehicle which created the Source Packet. (Note: the space vehicle itself is identified by the Spacecraft Identifier in the Transfer Frame header.) The Application Process IDs are tailored to local mission needs and are therefore assigned by the Mission Manager. Guidelines for assigning Application Process IDs may be developed by the CCSDS. Users should note that ground data accounting considerations may limit the number of different application processes which may be simultaneously "open" during a given session.

The "all ones" configuration of the Application Process ID shall be reserved to identify "Idle Packets", which are generated by the spacecraft data system to maintain synchronism of the ground packet extraction process during periods when no sources have packetized data available for transfer to the ground.

3.1.2 PACKET SEQUENCE CONTROL (16 Bits)

This 16-bit field is separated into two fields:

(1) Segmentation Flags (Bits 0,1)

The Segmentation Flags, which occupy the two most-significant bits of the 16-bit field, are used to indicate the status of long message-oriented Source Packets that have been broken into shorter communications-oriented segments. Various optional processes for segmentation are described in Section 4. The assignment of the flags is as follows:

(a) Last Segment Flag (Bit 0)

When Bit 0 is set to a "1", this indicates that the remainder of the data structure contains the last segment of a long segmented Source Packet.

(b) First Segment Flag (Bit 1)

When Bit 1 is set to a "1", this indicates that the remainder of the data structure contains the first segment of a long segmented Source Packet.

The Segmentation Flags are thus interpreted as follows:

Bit 0	Bit 1	Interpretation
0	0	Continuation Segment
0	1	First Segment
1	0	Last Segment
1	1	Unsegmented Packet

NOTE: When a source does not perform segmentation, it shall set the flags to "11" before passing the packet to the spacecraft data handling system for transfer to the ground.

(2) Source Sequence Count (Bits 2 through 15)

This 14-bit field contains a straight sequential count (modulo 16384) of each packet generated by each unique source application process on the spacecraft. The purpose of the field is to associate this packet with other packets from the same application process, even though their natural order may have been disturbed during transport to the user's processor on the ground. The field will normally be used in conjunction with the spacecraft measurement time code to provide completely unambiguous association; it is therefore essential that the period of the Source Sequence Count should be sufficiently long that the time code increments at least once between successive recycling of the sequence count. For continuous operation of a source application process, it is not permissible to "short cycle" the sequence count before the full counter accumulation has been reached; however, if a source's operation is interrupted (e.g., by loss of power), the source may start a new sequence count.

3.1.3 PACKET LENGTH (16 Bits)

This field contains a sequential 16-bit binary count "C" of the length (in octets) of the remainder of the data structure which is enclosed between the first bit of the Secondary Header and the last bit of the packet (i.e., the last bit of the Source Data field, or the last bit of the Error Control field if this option is selected). The field is expressed as follows:

C = { (Number of octets) - 1)

Users should recognize that although very long packets are permissible, these present special problems in terms of data link monopolization, source data buffering, and network accountability during transfer across the unique channel from the spacecraft to the ground. As described in Section 4, large packets (i.e., packets which are very much longer than the data space within the Transfer Frame) may either be segmented by the transfer network or may be assigned to individual "Virtual Channels". It is anticipated that for a large number of applications, packet lengths in the range of approximately 1 to 8 kilobits will be chosen by instrument designers for flight operations. Very short packets (less than'1024 bits) become inefficient and will normally only be used by very low-rate data sources, for engineering tests and checkout, and for fill.

For flight operations, computational speed considerations within the ground data processing elements of the transfer network may dictate that there is a maximum incoming packet rate which can be handled. This may mean that the network implementation organizations within each Agency will specify a minimum size for small packets (e.g., "Idle Packets") which are transmitted contiguously.

From the viewpoint of ground data processing efficiency, it is strongly recommended that the overall packet length should be an even number of octets.

3.2 SECONDARY HEADER

The purpose of the Secondary Header is to provide a CCSDS-defined means for encoding within a Source Packet any ancillary data (time, internal data field format, spacecraft position/attitude, etc.) which may be necessary for the interpretation of the information contained within the packet. The presence or absence of a Secondary Header is signalled by the Secondary Header Flag within the Packet Identification field, as described in Section 3.1.1(3). If present, the Secondary Header will contain data generated by one or more onboard application processes.

Short-term packet management processes within the ground network, such as extracting and delivering individual packets to a user in near-real time, may be able to use only the sequence count field within the packet header for accounting purposes. However, for longer-term services to be provided (e.g., archiving, sorting, processing and correlation with other data sets) the sequence count must be concatenated with a "time" field in order to unambiguously identify a packet. For users needing these long-term services, it is a requirement that the Secondary Header must be present in every packet produced by the source and must contain a time code which is registered with respect to some known event encoded within the Source Data section of the packet, accurate to the time resolution required for the Source Data interpretation and association.

The length of the Secondary Header shall always occur in integral multiples of octets. At present, the CCSDS has not developed a comprehensive recommendation for the format of the Secondary Header: therefore, local standards (defined by CCSDS Agencies, Centers, or Missions) may be applied as required. However, in order to permit future standardization of the Secondary Header, the most significant bit (bit 0) of the leading octet of the Secondary Header shall be set as follows:

Bit0 = "0" is defined to mean: "Non-CCSDS-defined Secondary Header follows".

Bit0 = "1" is defined to mean: "CCSDS-defined Secondary Header follows".

Processing of non-CCSDS-defined Secondary Headers (signalled by the Secondary Header flag being value “1” and bit 0 of the leading octet of the Secondary Header being value "0") by a supporting Agency shall be the subject of local agreements.

I

3.3 SOURCE DATA

The Source Data field contains the measurement information generated by the primary application process operating within each source. The only formal restriction imposed on the Source Data field is that the total length of this section must be an integral number of octets; otherwise, the experimenter will normally have complete freedom to specify the data content and the internal format of this field. However, users are cautioned that if the packet contents are to be processed within CCSDS Agency support facilities, then local standards for internal formatting may be imposed.

3.4 PACKET ERROR CONTROL (Optional)

At the discretion of the user, an optional error detection code may be appended to the packet in order to verify that the overall integrity of the message has been preserved during the transport process. The selection of the encoding polynomial, and the length of the field, is left to the user or to local Agency standards or agreements. The presence or absence of packet error control will be implied by the Application Process ID.

4 PACKET SEGMENTATION

The recommendations for Source Packet formatting permit a wide range of packet lengths to be implemented. Furthermore, sources may conceptually vary the packet length on a dynamic basis, according to the message formatting needs of different application processes operating within the source.

Space communications systems are usually heavily driven by the bandwidth or capacity constraints of the unique data channel which connects the spacecraft to the ground. Since multiple users must share this communications channel, flow control becomes critical to ensure that all sources have access to this common resource for periods of time consistent with their delivery timeliness requirements and their capacity to buffer data while other sources are being serviced. Very long Source Packets therefore present a flow control problem since they may monopolize the channel for unacceptable periods of time and may force other sources to implement unreasonably large local buffering.

Several mechanisms are provided for solving this flow control problem within Packet Telemetry systems. One involves assigning long-packet generating sources to their own "Virtual Channel" by inserting them into dedicated types of Transfer Frames. These dedicated frames may then be interleaved with other frames containing multi-user data which are formatted into mutually compatible packet sizes. This Virtual Channel solution is discussed in Section 5.

Another mechanism involves the use of various "segmentation" protocols whereby the spacecraft data handling system (or the source itself) breaks the long packets into shorter pieces which are compatible with the flow control requirements of other users. The long-packet source will therefore either be responsible for performing its own communications-oriented segmentation or for buffering its own data while the spacecraft breaks the packet up into smaller segments and interleaves them with similar-sized segments of data from other sources for transfer to the ground.

This issue of the Packet Telemetry Recommendation permits the system designer to select between three valid methods of segmentation:

(1) Segmentation by the source application process, using the standard Version 1 "Source Packet" data structure (Section 4.1).

(2) Segmentation by the spacecraft data system, using the standard Version 1 "Source Packet" data structure (Section 4.2).

(3) Segmentation by the spacecraft data system, using the standard Version 2 "Telemetry Segment" data structure (Section 4.3).

These three methods are discussed below. Virtual Channelization is discussed in Section 5: The attributes of all four methods are summarized in Reference [4]. Selection of the specific option(s) which will be cross-supported by CCSDS Agencies will be defined within the detailed cross-support agreements.

4.1 SOURCE-INTERNAL SEGMENTATION

A first option is for the source application process to be constrained by the system designer to always generate packets which are short enough to ensure that no flow control problems can exist when they are integrated with packets from other onboard sources; however, this may artificially constrain the measurement acquisition processes by locking them to the communication processes. For instance, an imaging scan line may be a "natural" packet for an instrument, but it may be too long to transmit unsegmented.

A second option is to allow the source application process to format measurement data into very long packets in accordance with sampling needs, but to require that the source interface electronics breaks them into shorter segments prior to delivery to the spacecraft data system. This may be accomplished by having the source reformat the long Version 1 packets into shorter Version 1 packets within which the Segmentation Flags (Section 3.1.2(1)) are manipulated to signal a first, continuing, or last segment of the larger packet entity. Within this option, the "Source Sequence Count" field must increment once for each segment generated and the "Packet Length" field must directly indicate the length of the segment; therefore, information which describes the Sequence Count and Length of the original large packet should be encoded within the data field of at least the first segment.

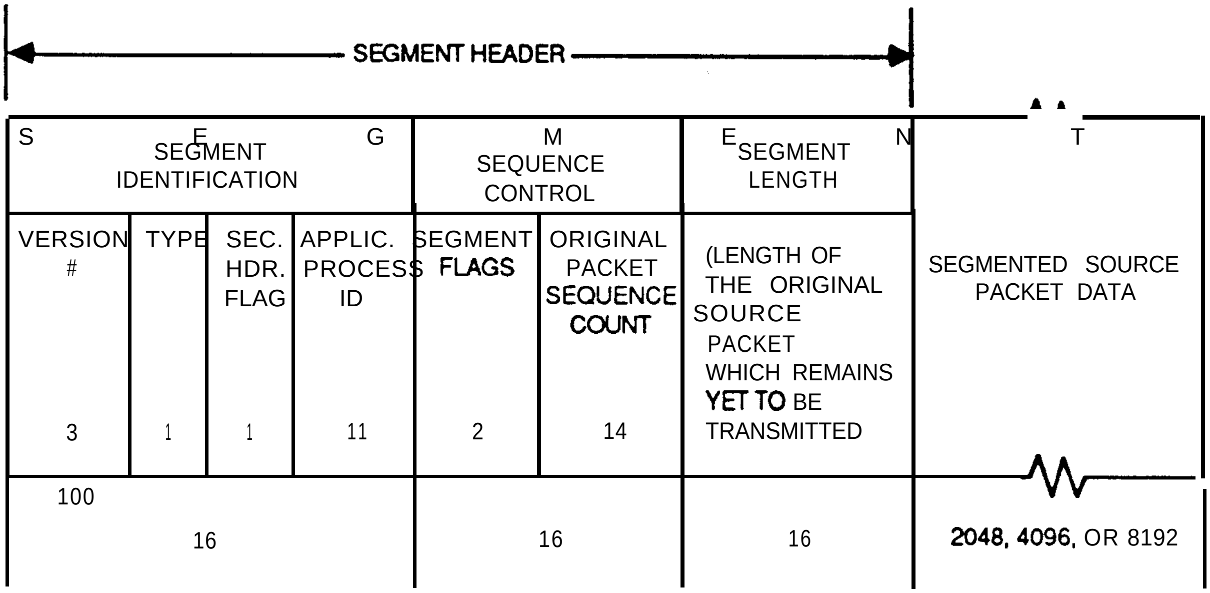
4.2 SPACECRAFT SEGMENTATION USING SOURCE PACKETS

A third option exists whereby the spacecraft data system implements one or more special internal application processes which are dedicated to the task of segmenting long packets from onboard sources. Within this option, the entire user packet is recursively treated as "data" in a new Version 1 packet, which bears Packet Identification, Packet Sequence Control and Packet Length fields that are unique to the spacecraft data system application process which performs the segmentation. In this case, the Segmentation Flags (Section 3.1.2(1)) of the new packet shall be manipulated to signal a first, continuing or last segment of the original user packet. This solution introduces a modest amount of extra overhead, since the first segment in such a system will contain the Primary Header of the spacecraft data system segmentation process, followed by the complete Primary Header of the packet which is being segmented.

4.3 SPACECRAFT SEGMENTATION USING "TELEMETRY SEGMENTS"

All three of the above options for segmentation utilize the Version 1 Source Packet format, within which the "length" field is always interpreted as the length in octets of the data unit (packet or segment) that appears on the downlink and the "sequence count" increments once per data unit that is transmitted from a given application process. A **fourth option** exists, **using** "Version 2" of the packet format (see **Section 3.1.1(1)),** within **which the "kngth" field in the** data unit defines the length of an original packet that remains to be transmitted and the "sequence count" is static since it refers **to the** numbering of the original packet by the application process. The length and sequence count of the data unit being transmitted are therefore semantically different **between the two versions, since these parameters are implied when using Version 2.**

The Version 2 "Telemetry Segment" structure **is shown in Figure 4-l.**



SEGMENT HEADER

S E G M E N T

SEGMENT IDENTIFICATION				SEQUENCE CONTROL		SEGMENT LENGTH	
VERSION #	TYPE	SEC. HDR. FLAG	APPLIC. PROCESS ID	SEGMENT FLAGS	ORIGINAL PACKET SEQUENCE COUNT	(LENGTH OF THE ORIGINAL SOURCE PACKET WHICH REMAINS YET TO BE TRANSMITTED	SEGMENTED SOURCE PACKET DATA
3	1	1	11	2	14		
100							
16				16		16	2048, 4096, OR 8192

Figure 4-1: Version 2 "Telemetry Segment" Format

The semantic definition of the Version 2 "Segment Header" differs from the Version 1 (Source Packet) "Primary Header" in two important ways:

(1) The Packet Sequence Count field is static within all the segments associated with a long packet which is being segmented since it contains the count of the original packet.

(2) The Segment Length field does not indicate the length of the segment, but instead indicates the length of data from the original long packet (including that contained within the segment) which remains yet to he transmitted. The length of the segment is fixed and is specified externally.

The length of the segment of data from the original Source Packet which is being transmitted **("LSEGMENT",** in octets) shall be fixed for a particular Virtual

Channel on ‘a given mission. The available fixed lengths are LSEGMENT = 256, 512 or 1024 octets (2048.4096 or 8192 bits), with 512 octets being the preferred value since shorter lengths result in excessive overhead and ground computational speed requirements. The selected length shall be indicated in the header of the Transfer Frame, as described in Section 5.2.4(d).

Note that the total length of each Telemetry Segment will be (LSEGMENT+6) octets since the standard 48-bit Version 2 Segment Header always precedes the segment of data.

4.3.1 SEGMENTATION PROCESS

When using the Version 2 Telemetry Segment to perform segmentation, the process shall be as follows:

(1) An unsegmented Version 1 Source Packet is input to the processor which performs the segmentation.

(2) In the first segment, the Version Number is modified to indicate Version 2 (i.e., "100"), and the Segmentation Flags (Bits 0 and 1 of the Packet Sequence Control field) are modified from the "11" state (unsegmented) to the “01” state (first segment), as described in Section 3.1.2(1). In the first segment, the Packet Length field is unchanged and therefore indicates the length of the original packet WHICH HAS YET TO BE TRANSMITTED, including this segment. This field is then followed by the first (LSEGMENT) octets of the Secondary Header and Data Field of the original Version 1 packet.

(3) In continuation segments, the Version Number continues to indicate Version 2, the Segmentation Flags are set to the "00" configuration and the original Packet Length field is progressively decremented by (n) x (LSEGMENT) octets as each segment is transmitted, where (n) = 1,2,3, etc. according to the sequential number of the continuation segment.

(4) In the last segment, which occurs when the original Packet Length field decremented by (n) x (LSEGMENT) octets contains a value which is less than or equal to (LSEGMENT), the Version Number remains as Version 2, the Segmentation Flags are set to "10", and the Packet Length field then directly indicates the length of the residue of the original packet which is contained within the segment.

This process is illustrated in Figure 4-2, which shows how a Version 1 Source Packet whose overall length is 2106 octets (2100 octets of Secondary Header and Source Data plus 6 octets of Primary Header) is progressively broken into four 5 12-octet Version 2 Telemetry Segments of overall length 518 octets (512 octets of data plus 6 octets of Segment Header) and a last segment of overall length 58 octets (52 octets of data plus 6 octets of Segment Header). Note that the convention of:

"Indicated length = { (Number of octets) - 1 }"

is used for the Segment Length field, as defined in Section 3.1.3.

It should also be noted that it is permissible to generate Version 2 Telemetry Segments which have the Segmentation Flags set to "11" (unsegmented): in this case the Segment Length field indicates the length of the remainder of the Segment.

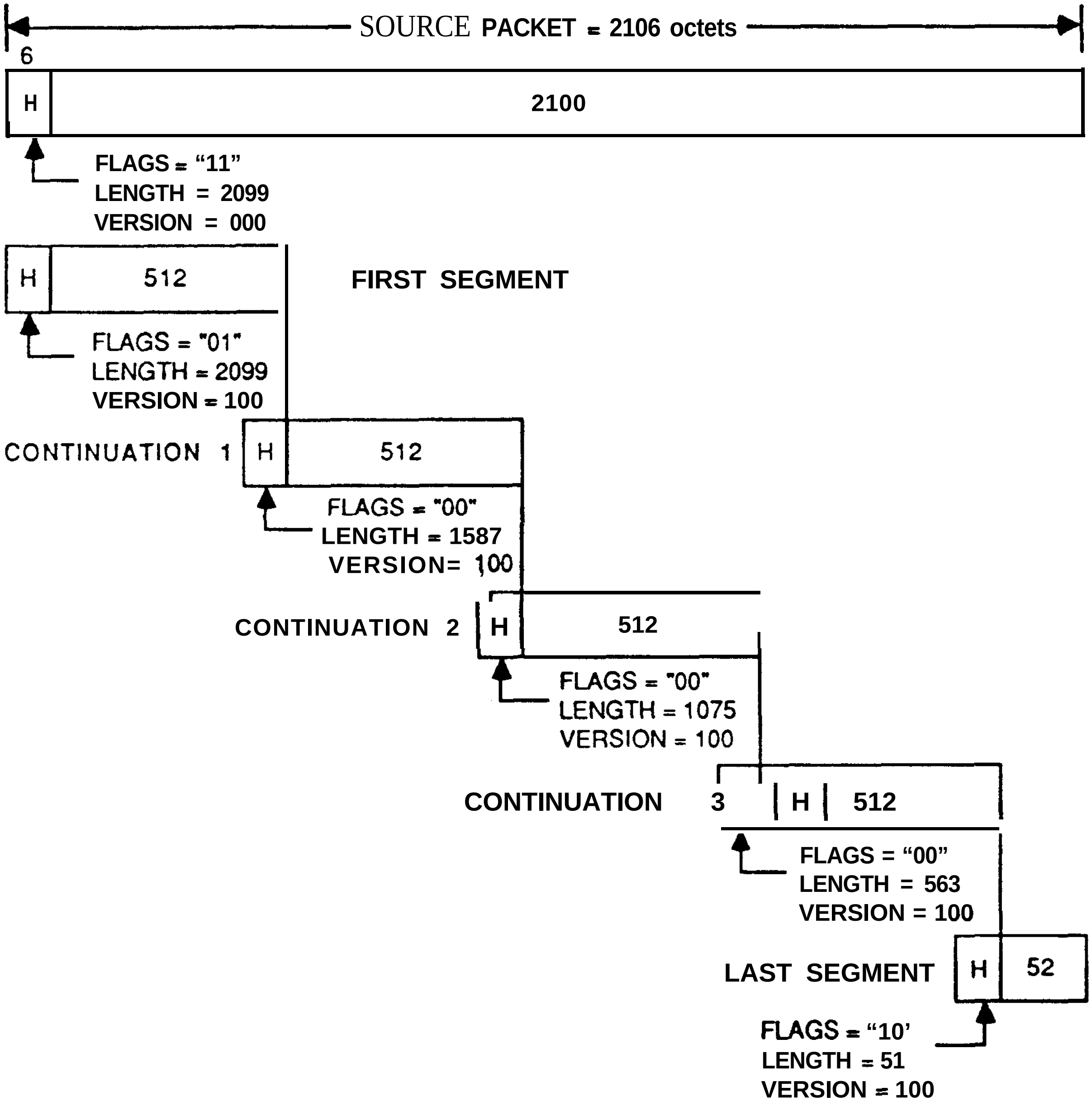


Figure 4-2: Example of the Segmentation Process

Attention is drawn to the following features of this protocol:

(1) It is possible to mix segmented and unsegmented Source Packets together on the same Virtual Channel, since every packet header contains a Version Number and self-specifying Segmentation Flags.

(2) Within the ground processor which extracts Version 2 segments from the frame, the location of the Segment Headers may be determined from a simple test of the Last Segment flag in the Segment Sequence Control field:

(a) If the Last Segment flag is "0", then the next header will be found by counting forward by (LSEGMENT) octets after the end of the Segment Length field.

(b) If the Last Segment flag is "1", then the next header will be found by counting forward by the number of Octets indicated in the Segment Length field.

(3) Since the fixed segment lengths are defined to be pure binary quantities (256,512 or 1024 octets), then by implementing this decrementing length approach the most significant bits of the Segment Length field will decrease in a binary countdown fashion as successive segments are transmitted. This information provides a "serial number" for the segment which may be used to recombine segments should their natural order be disturbed during transmission. For example, with 512-octet segments, the most significant octet of the length field will form this binary down-counter. Using the same case which is shown in Figure 4-2, the Length field thus would appear as follows:

LENGTH BIT:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
1 ST SEGMENT:	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1
CONTINUATION-1 :	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
CONTINUATION-2:	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1
CONTINUATION-3:	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1
LAST SEGMENT:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

COUNTER
(= SEGMENT
SERIAL #)

5 TRANSFER FRAME FORMAT

The Version 1 Source Packet or Version 2 Telemetry Segment data formats described in Sections 3 and 4 must be embedded within a data transfer structure for transmission across the downlink data channel which connects the spacecraft to a data capture element on the ground: this data structure is the "Transfer Frame". Multiple CCSDS standard versions of the Transfer Frame may be defined in the future; however, this issue of the Recommendation for Packet Telemetry only recognizes "Version 1" of the Transfer Frame format.

The Transfer Frame draws upon a layer of noisy channel services (e.g., carrier, modulation/detection, and coding/decoding) in order to establish the downlink data path. The error probabilities attainable within the noisy channel layer depend upon many factors, including received signal-to-noise ratio and coding scheme used. Digital encoding may be applied to the data channel in order to improve the system error performance. Although Packet Telemetry systems may be designed to tolerate channel noise in the same way that conventional systems have been designed in the past (i.e., by placing data within the frame in a predetermined sequence), full benefit from Packet Telemetry will require that a high quality data channel is provided so that packetized data may be ADAPTIVELY inserted into the frame. Reference [2] describes the CCSDS Recommendation for Telemetry Channel Coding, including specification of a convolutionally encoded inner channel concatenated with a Reed-Solomon block-oriented outer code. Although not mandatory, the recommended approach is to use the concatenated Reed-Solomon/convolutional encoding option since the data channel through which the frame is transmitted will then display virtually perfect data quality.

A mechanism must be provided to detect the presence of errors which may have been introduced within the frame during transmission through the downlink channel. If the recommended concatenated Reed-Solomon/convolutional coding is used, the Reed-Solomon error syndrome will indicate whether or not the frame is likely to contain an error. If Reed-Solomon encoding is NOT used, each Version 1 Transfer Frame must contain an error-detecting polynomial appended within its trailing octets: this polynomial may be used to determine if a frame is likely to contain an error.

The operational procedures for handling frames which contain a detected error are beyond the scope of this Recommendation and will therefore be negotiated between the Agencies when detailed cross-support agreements are formalized This Recommendation also recognizes that Projects may desire to implement special "emergency" telemetry modes for the transmission of critical information in the event of potentially catastrophic behavior, such as the loss of spacecraft attitude references. If these emergency modes are beyond the scope of this Recommendation, they will also be the subject of detailed cross-support agreements.

Within a single physical data channel for a particular mission, the total length of the Version 1 Transfer Frame shall be a fixed, integral number of octets: the implemented length shall be specified to the ground network as a mission set-up parameter. Although each Agency may set specifications for the minimum length of the Version 1 Transfer Frame to be used on missions under its control, it is desirable to select a common maximum value for cross-support situations in order to avoid unnecessary variability and complexity. The maximum'length has been computed for the case where the Transfer Frame is synchronously inserted into a Reed-Solomon codeblock structure. For cross-support situations, a Reed-Solomon interleave depth of five is recommended. Because the data space within this recommended (I = 5) Reed-Solomon codeblock has a fixed maximum length of 8920 bits, a standard maximum Version 1 Transfer Frame length of 8920 bits (not including the 32-bit attached sync marker) has been selected for missions which are cross-supported between CCSDS Agencies since this is compatible with synchronous insertion of the Frame into the codeblock data space.

Figure 5-l illustrates how a maximum length Version 1 Transfer Frame may be synchronously embedded in the standard Reed-Solomon codeblock

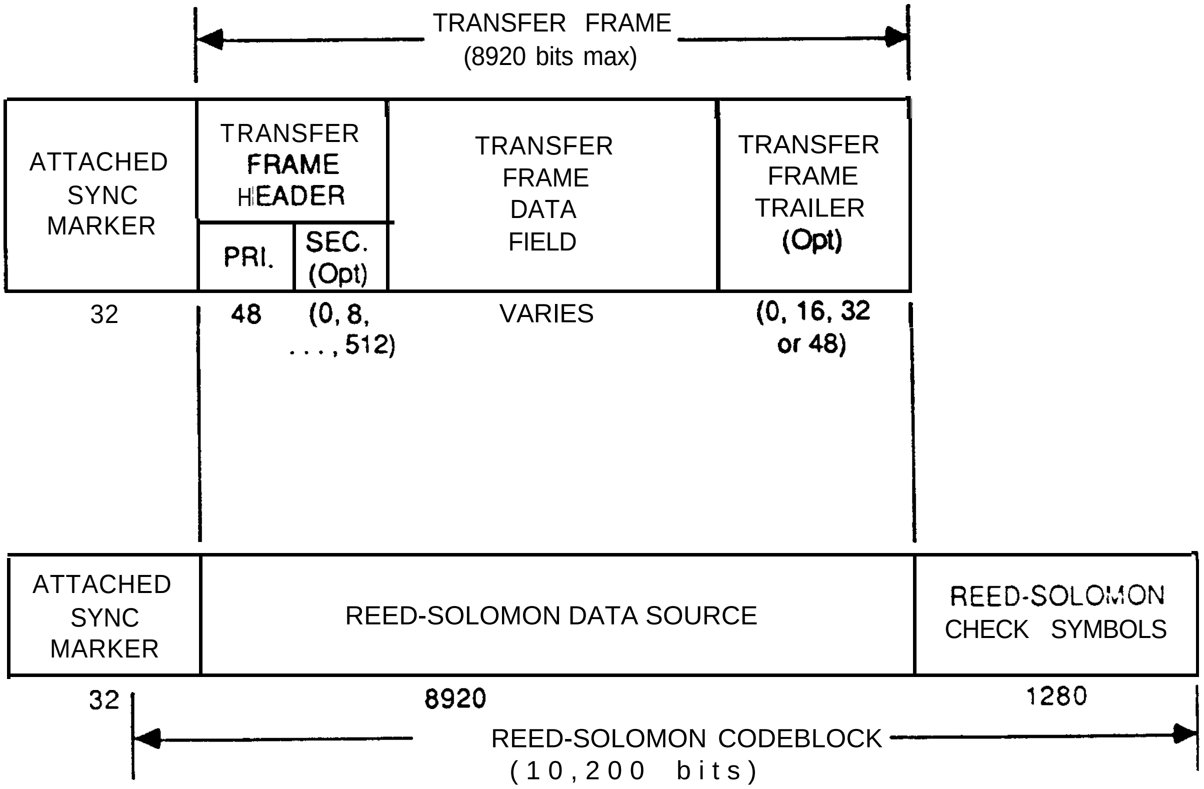


Figure 5-1: Composite Version 1 Transfer **Frame/ Reed-Solomon Codeblock Format**

The major fields of the composite Version 1 data structure shown in Figure 5-l are as follows:

Major Field	Length (Bits)	
ATTACHED 32-BIT FRAME SYNCHRONIZATION MARKER	(Not incl.)	(Note 1)
TRANSFER FRAME PRIMARY HEADER	48	
TRANSFER FRAME SECONDARY HEADER (Optional)	(0,8,16...512)	
TRANSFER FRAME DATA FIELD	(Varies)	(Note 2)
TRANSFER FRAME TRAILER (Optional)	(0,16,32, or 48)	
Total Length:	Up to 8920	(Note 2)

NOTES:

1. As a transitional measure for the cross-support of ongoing missions, a 16-bit marker may be implemented if short frames are transmitted without Reed-Solomon coding, through a channel which displays good received signal-to-noise ratio characteristics.

2. The maximum frame length for cross-support is 8920 bits, which consists of: the mandatory Primary Header (48 bits); the optional Secondary Header (up to 5 12 bits); the Frame Data Field; and the optional Frame Trailer (16, 32 or 48 bits). The maximum length of the Data Field is therefore computed by subtracting the selected mandatory and optional components from 8920 bits. The 32-bit Attached Sync Marker is added to the frame length to determine the length upon which the frame synchronizer operates. This marker is not part of the transfer frame in order to avoid the possibility of including it within a Reed-Solomon codeblock. Caution: Projects which implement frame lengths shorter than the maximum, and which insert these frames synchronously into the chosen (I = 5) Reed-Solomon data space, must ensure that the correct amount of “Virtual Fill” is inferred by the ground decoder, as described in Reference [2].

5.1 ATTACHED FRAME SYNCHRONIZATION MARKER

The Attached Frame Synchronization Marker delimits the boundaries of a fixed-length Transfer Frame. If the frame is NOT Reed-Solomon encoded, it is used by ground equipment to acquire synchronization with the frame boundaries after transmission through the data channel. If the frame IS Reed-Solomon encoded, the marker serves to synchronize the Reed-Solomon codeblock. Rules for these two cases of attaching the Synchronization Marker to the Transfer Frame or to the Reed-Solomon codeblock are described in the next sections.

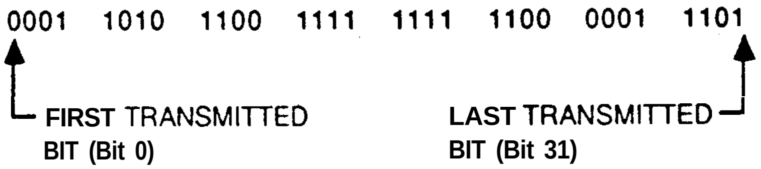
All cross-supported missions using a packet telemetry format shall use the same frame synchronization marker to avoid the need to set up the ground-based PCM frame synchronizers in advance of every session. Care should be taken that this synchronization marker (or its reversed and/or complemented pattern) does not routinely appear in any other portion of the

Transfer Frame. This does not preclude the occasional random presence'of this pattern elsewhere in the frame.

5.1.1 SYNCHRONIZATION MARKER WITHOUT REED-SOLOMON CODING

When Reed-Solomon coding is NOT used as part of the overall channel code, all Transfer Frames shall have a 32-bit synchronization marker attached to and immediately preceding the Transfer Frame Header, as shown in Figure S-l. After frame synchronization has been performed on the ground, the marker shall remain attached to the beginning of the frame.

When Reed-Solomon Coding is NOT used, the recommended 32-bit synchronization marker pattern is as follows:



This pattern may be represented in hexadecimal notation as:

1ACFFC1D

This pattern was chosen (Reference [4]) to provide good synchronization properties with a low false alarm probability in a noisy channel under the following conditions: True as well as complemented data sense, forward as well as reverse time ordering, and synchronization directly in the bit domain as well as in the symbol domain (as translated by the recommended convolutional code of Reference [2].)

5.1.2 SYNCHRONIZATION MARKER WITH REED-SOLOMON CODING

When Reed-Solomon coding is used as part of the overall channel code, and each Transfer Frame is synchronously imbedded within a Reed-Solomon codeblock, the codeblock shall have an Attached Sync Marker immediately preceding it, as described in Reference [2]. It should be noted that this marker is NOT part of the codeblock's 8920-bit data space, and that the imbedded Transfer Frame should NOT include its own marker. After decoding, the marker shall be reattached to the beginning of the decoded codeblock. Since synchronization of the codeblock on the mound must first be achieved for the decoding process to begin (before the Transfer Frame is-extracted), the bit pattern for the Attached Synchronization Marker of the codeblock is specified and controlled by Reference [2].

5.2 TRANSFER FRAME PRIMARY HEADER

The Transfer Frame Header is split into “Primary” and “Secondary” elements.‘ The mandatory Primary Header provides services which are common to all missions. The optional Secondary Header, which is described in Section 5.3, permits the header information to be tailored to match the needs of different mission classes. The composite Transfer Frame structure, containing the Primary Header and Secondary Headers, plus the Frame Data Field and optional Frame Trailer, is shown in Figure 5-2.

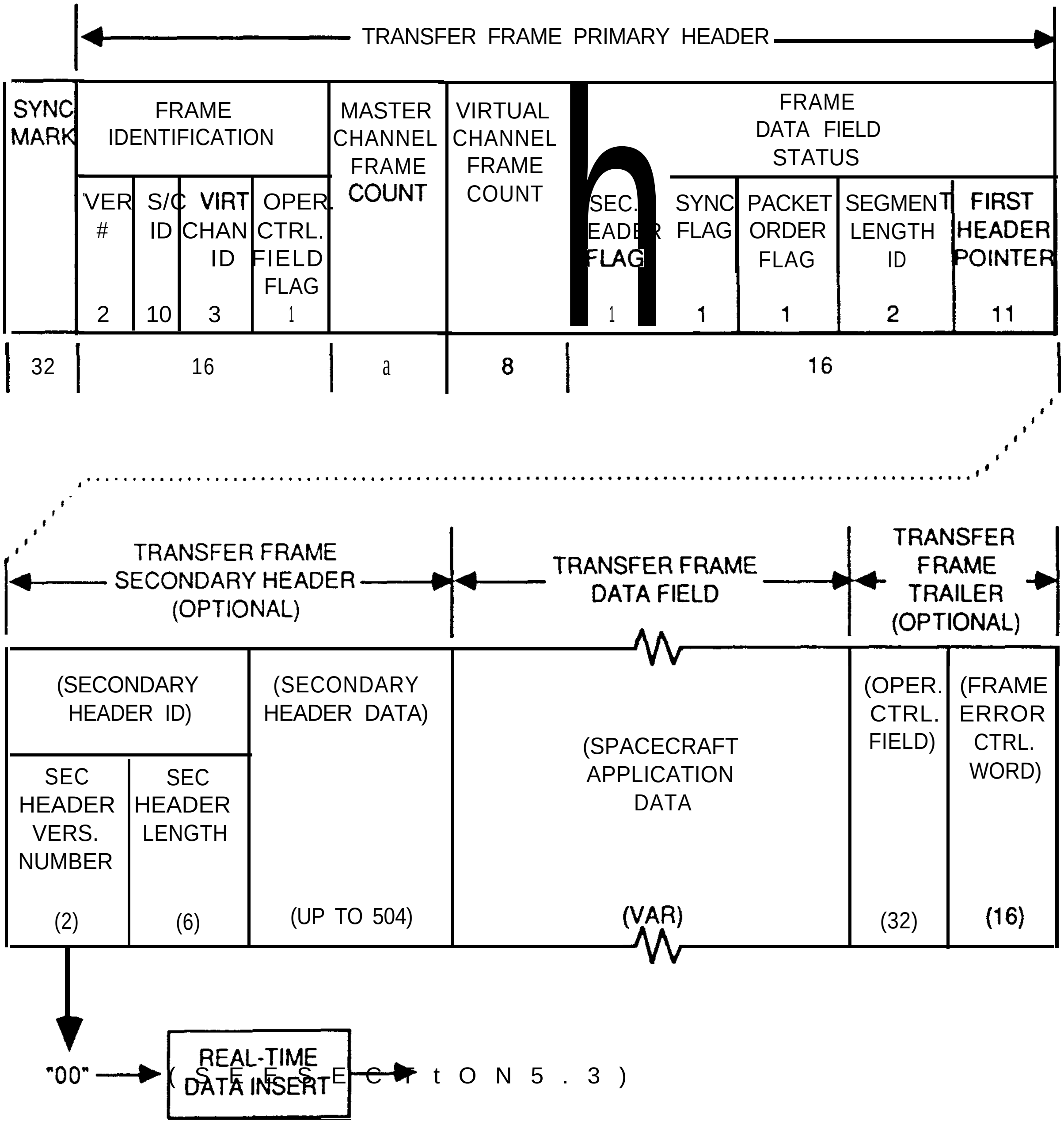


Figure 5-2: Version 1 Transfer Frame Header Format

The Primary Header performs four principal functions:

(1) Identifying the Version of the Transfer Frame which is in use.

(2) Identifying the spacecraft which transmitted the telemetered data.

(3) Switching of the single physical data channel so that it may be logically multiplexed into several "Virtual Channels", and providing accounting mechanisms to detect missing frames.

(4) Providing pointers and other control information so that variable-length Source Packet or Segment data may be extracted from the fixed-length Frame Data Field.

The VERSION 1 Transfer Frame Primary Header is composed of the following fields:

Major Field	**Length (Bits)**
Frame Identification	**16**
- Version Number (2)	
- Spacecraft ID (10)	
- Virtual Channel ID (3)	
- Operational Control Field Flag (1)	
Master Channel Frame Count	8
Virtual Channel Frame Count	8
Frame Data Field Status	16
- Secondary Header Flag (1)	
- Synchronization flag (1)	
- Packet Order flag (1)	
- Segment Length ID (2)	
- First Header Pointer (11)	
	48

5.2.1 FRAME IDENTIFICATION (16 Bits)

The purpose of this field is to identify which operational spacecraft created the frame of data, and to indicate if the physical data link is logically switched to form "Virtual Channels". The field is broken into four subfields:

(a) Version Number (Bits 0,1)

These two bits (which occupy the two most significant bits of the 16-bit field) are reserved for potential evolution of the Transfer Frame structure. One possible future use of this capability could be to extend the Spacecraft ID word if all ten bits become assigned. **At present, only Version 1 of the Frame Header is**

recognized (Bits 0,1 = "00"). The format of the remainder of the Version 1 header is as follows:

(b) Spacecraft Identifier (Bits 2 through 11)

These ten bits provide positive identification of the spacecraft node which created the frame of data. Different spacecraft identifiers will be assigned for flight vehicles, for developmental vehicles which are using the ground networks during prelaunch test operations, and for simulated data streams. The Secretariat of the Consultative Committee for Space Data Systems assigns spacecraft identifiers, as described in Reference [1].

(c) Virtual Channel Identifier (Bits 12,13,14)

The concept of using a “Virtual Channel” as a solution to the problem of long Source Packet segmentation was outlined in the introduction to Section 4. The concept may be implemented by defining different logical “types” of Transfer Frames, each of which is separately identified as a different Virtual Channel. Long Source Packets may thus be inserted only into their own dedicated "L" frame type, while shorter multi-user packets may be multiplexed together into a different "M" frame type. The composite physical stream of downlink Transfer Frames may then be created by interleaving logical frames with different "L" and "M" Virtual Channel identifiers. The effect on the long packet generating source will be similar to the Segmentation process described in Section 4, i.e., the packet will be broken into fixed blocks, the length of which will equal the length of the Data Field within the Transfer Frame. During times when the other source data are being transmitted within their "M" Type of frame, the long packet source data will be buffered until an opportunity occurs for the spacecraft to interleave an "L" Type frame containing a segment of that packet.

The Virtual Channel facility also allows complete frames of data from other spacecraft generating sources (e.g., tape recorder playback, relay links from other spacecraft) to be interleaved with real-time frames.

This 3-bit field enables up to eight “Virtual Channels” to be run concurrently by a particular spacecraft on a particular physical data channel. The sequence in which Virtual Channels are multiplexed is mission-dependent. If only one Virtual Channel is used, these bits shall be set permanently to "000".

(d) Operational Control Field Flag (Bit 15)

This l-bit flag signals the presence (Bit 15 = "1") or absence (Bit 15 = "0") of the 32-bit Operational Control Field within the Frame Trailer (see Section 5.5).

5.2.2 MASTER CHANNEL FRAME COUNT (8 Bits)

The purpose of this field is to provide a running count of the number of frames which have been transmitted through a single spacecraft physical data channel. Some spacecraft may have the capability to create more than one physical data channel to the ground, in which case a separate counter will therefore be maintained for each channel. The counter must be long enough to provide a reasonable probability of detecting how many frames were missing if the physical channel is briefly interrupted. The 8-bit field represents a sequential count (modulo 256) of each Transfer Frame generated by the spacecraft on a given physical data channel.

5.2.3 VIRTUAL CHANNEL FRAME COUNT (8 Bits)

The purpose of this field is to provide individual accountability for each of the eight “Virtual Channels”. The 8-bit field represents a sequential count (modulo 256) of the total number of frames which have been transmitted in association with EACH of the virtual channels. It is used in association with the “Virtual Channel ID” field to maintain a separate counter for each of up to eight separate virtual channels.

5.2.4 FRAME DATA FIELD STATUS (16 Bits)

The purpose of this field is to provide control information necessary to enable Packets or Segments to be extracted from the Frame Data Field. The field is broken into the following subfields:

(a) Secondary Header Flag (Bit 0)

This l-bit flag indicates the presence (Bit 0 = "1") or absence (Bit 0 = "0") of the optional Secondary Header in the Transfer Frame. If present, the Secondary Header shall immediately follow the Primary Header and the beginning of the Frame Data Field shall be correspondingly shifted.

If implemented, the Secondary Header shall appear in every frame transmitted through a physical data channel, and its length shall be fixed within that channel.

(b) Data Field Synchronization Flag (Bit 1)

The normal mode of inserting Packet/Segment data units into the Frame Data Field shall be to synchronously place them on octet boundaries so that they follow directly after each other. The Packets or Segments will thus be permitted to “spill over” into the next frame, and the location of the FIRST Packet/Segment header in a particular frame will be specified by the First Header Pointer field. For this synchronous insertion mode, the Data Field Synchronization Flag shall be set to a "0".

If a Project chooses to place Packet/Segment data ASYNCHRONOUSLY within the frame data field so that octet boundaries are not observed, then the Data Field Synchronization Flag within that frame shall be set to a "1". This situation may occur if the Project elects not to achieve onboard Packet/Segment bit rejustification when a tape recorder is replayed in reverse (thus dumping an unstructured bit stream into the data field), or if the source data do not otherwise conform to the Packet/Segment protocols. IF THIS FLAG IS SET TO A " 1 ", INDICATING ASYNCHRONOUS DATA INSERTION, THE REMAINING FRAME DATA FIELD STATUS INFORMATION MAY NOT BE VALID, AND IT SHALL BE A PROJECT RESPONSIBILITY TO EXTRACT INFORMATION FROM THE FRAME DATA FIELD.

The synchronization status of the Packet/Segment data which are inserted into the Frame Data Field shall be indicated by setting the Data Field Synchronization Flag (Bit 1) as follows:

Bit 1 = 0 : Packet/Segment data are synchronously inserted.
Bit 1 = 1 : Packet/Segment data are asynchronously inserted.

(c) Packet Order Flag (Bit 2)

During normal real-time transmission of information from spacecraft sources to the ground, the Source Packets or Telemetry Segments inserted within the Frame Data Field (as they appear to the ground processor) will be "forward" justified, i.e., they will appear with their most-significant bit transmitted first, and with their sequence counters incrementing in an increasing order. The Packet Order Flag shall be used to indicate certain conditions where the ORDER of the sequence counters within the Packets or Segments may be reversed.

When contact with a ground station is not maintained, or when the transmission capacity of the downlink channel is less than the demands of the data sources, the spacecraft may record telemetry data on an onboard storage device. If this device is a tape recorder, then this Recommendation recognizes that it may be desirable to replay the recorder in a reverse direction, causing the order of the transmitted data to be reversed.

For CCSDS cross-supported missions the baseline requirement is that the spacecraft shall re-justify the BIT DIRECTION of any Packet/Segment data which have been replayed in reverse, prior to inserting them into the Transfer Frame, so that the most significant bit is transmitted first. This will require Packet/Segment synchronization logic at the output of the tape recorder to reverse the bit direction of each Packet/Segment as it is retrieved. Under these conditions, the ORDER of the Packet/Segment sequence counters will be observed to decrease rather than increase.

(NOTE: If replayed Packets/Segments appear in reverse order, they must be multiplexed together on a separate Virtual Channel: it is not permissible to mix forward-ordered and reverse-ordered Packets/Segments within the same Virtual Channel since the Packet Order Flag would be ambiguous.)

The ORDER of the sequence counts within the Packet/Segment data contained within the frame data field shall be indicated by setting the Packet Order Flag (Bit 2) as follows:

Bit 2 = 0 : Packet/Segment sequence count order is "forward".
Bit 2 = 1 : Packet/Segment sequence count order is "reverse".

A discussion of various options for handling tape recorded data is contained in Reference [4].

(d) Segment Length Identifier (Bits 3,4)

As discussed in Section 4, Version 2 Telemetry Segments may be implemented within a Virtual Channel as a method of controlling the flow of data from sources which generated very long packets. This 2-bit field identifies the selected maximum length of the standard Version 2 Telemetry Segment, if this option is being used within the Virtual Channel which is formed by the frame. The contents of the field are interpreted as follows:

00 = 256-octet segment
01 = 512-octet segment
10 = 1024-octet segment

If Version 2 Telemetry Segments and Version 1 Source Packets are mixed within the same Virtual Channel, the Version 1 packets may not be longer than the indicated maximum segment length. If the Version 2 Telemetry Segment is not being used within the Virtual Channel, the field shall be set to the "11" state.

(e) First Header Pointer (Bits 5 through 15)

Packet or Segment headers shall be aligned with octet boundaries within the Frame Data Field. The purpose of this field is to point directly to the location of the starting octet of the first Packet or Segment header structure. The location of any subsequent headers within the same Frame Data Field will be determined by a "chaining" procedure whereby the Packet Length field (Section 3.1.3) within each Packet/Segment header structure will be examined to determine where the following header begins.

This 1 l-bit field contains a binary count "P" (modulo 2048) which, when incremented by "1", points directly to the number of the octet within the Frame Data Field (STARTING AT OCTET #"1", WHICH BEGINS AT THE FIRST

BIT OF THE FRAME DATA FIELD) that contains the first octet of the first Packet or Segment header structure. The count "P" is expressed as follows:

$$P = \{ \text{(Number of the octet)} - 1 \}$$

If the frame DOES NOT contain a Transfer Frame Secondary Header, the first bit of Octet # 1 within the Frame Data Field occurs immediately after the last bit of the Primary Header.

If the frame DOES contain a Secondary Header, then the first bit of Octet # 1 within the Frame Data Field occurs immediately after the last bit of the Secondary Header, i.e., it is offset by the length of the Secondary Header, which is specified within the Secondary Header Identification field. The value of the First Header Pointer is not affected by the existence of a Secondary Header.

If no Packet or Segment header structure starts in the Data Field, the First Header Pointer shall be set to " 11111111111" ("all ones"). This situation may occur if a long packet is segmented using the Virtual Channel technique.

If a Virtual Channel does not contain any. valid Packet or Segment data, the First Header Pointer shall be set to "11111111110" ("all ones minus one"). This may be used to signal an "Idle Channel" if no real or fill packets are available for transmission within the frame.

Since Packets or Segments may begin at any point within the Frame Data Field, it is possible that a Packet/Segment header may be split between successive frames. The rules for handling this situation are as follows:

(i) If the FIRST Packet/Segment header starts at the end of the Data Field within frame (N) and spills over into frame (N+1), the First Header Pointer in frame (N) shall indicate the start of this Packet/Segment header.

(ii) If ANY Packet/Segment header is split between frames (N) and (N+1), the pointer in frame (N+1) shall ignore the residue of the split header and shall only indicate the start of any subsequent new Packet/Segment header within frame (N+ 1).

5.3 TRANSFER FRAME SECONDARY HEADER (Optional)

The Transfer Frame Secondary Header is optional: its presence or absence is indicated by the Secondary Header Flag within the Primary Header. If implemented, the Secondary Header must be of fixed length and must appear in every frame transmitted through a physical data channel. Every Secondary Header shall begin with a single octet containing the "Secondary Header Identification", and shall have the general format indicated in Figure 5-2.

5.3.1 SECONDARY HEADER IDENTIFICATION (8 Bits)

This field defines the version and length of the Secondary Header: it is mandatory if a Secondary Header is present. It is separated into two subfields:

(a) Secondary Header Version Number (Bits 0,1)

This 2-bit subfield shall indicate which of up to. four Secondary Header versions is being used. By changing the Version Number, future variations of the Secondary Header structure become possible. At present, only Version 1 (Bits 0,1 = "00") is recognized, and all other versions are reserved for future application. Within Version 1, the remainder of the Secondary Header format is defined as follows:

(b) Secondary Header Length (Bits 2 through 7)

This 6-bit subfield shall contain a binary count "S" of the total number of octets contained within the entire Transfer Frame Secondary Header (including the Secondary Header Identification field itself). The count "S" is expressed as follows:

S = ((Total number of octets) - 1) }

When a Secondary Header is present, this count consequently may be used to compute the number of octet boundaries by which the first bit of the Frame Data Field is offset from the last bit of the Primary Header.

5.3.2 SECONDARY HEADER DATA (n x 8 Bits)

For Version 1 of the Secondary Header, this field (which must contain an integral number of octets and can be up to 63 octets long) contains a “Real Time Data Insert” of information that is required for various spacecraft monitoring and control applications. Guidelines for implementing the Real Time Data Insert are currently under development.

5.4 TRANSFER FRAME DATA FIELD

This field, which must exist as an integer number of octets, contains user application data (e.g., Packets or Segments) to be transferred from the spacecraft to the ground.

Packets or Segments shall be inserted contiguously into the Data Field on octet boundaries, with the location of the octet containing the first header being indicated by the First Header Pointer in the frame header. Subsequent headers are located by examining the “length” field in each packet, or by counting forward by the fixed Telemetry Segment length, whichever is applicable.

When there are no data available for transmission from any source, an “Idle Packet” may be inserted by the spacecraft data system for the purpose of keeping the frame running synchronously. The Idle Packet shall have the format of a Version 1 Source -Packet or a Version 2 Telemetry Segment, its Application Process ID shall read “all ones”, and its length may equal the minimum-available packet size.

The maximum length of the Frame Data Field depends on whether the optional Transfer Frame Secondary Header and Transfer Frame Trailer fields are present. As discussed in Reference [2], if frame lengths shorter than the 8920-bit maximum (not counting the 32-bit attached sync marker) are implemented and the frame is encoded using the recommended Reed-Solomon algorithm, then the length of the Frame Data Field must be selected bearing in mind the constraint that “Virtual Fill” must occur in fixed increments.

5.5 TRANSFER FRAME TRAILER (Optional)

The Transfer Frame Trailer provides a mechanism for inserting the following optional information into the trailing octets of the frame:

(1) An Operational Control Field, which facilitates closed-loop reporting of certain standardized real-time activities.

(2) A Frame Error Control Word, which facilitates detection of errors which may have occurred within the frame.

5.5.1 OPERATIONAL CONTROL FIELD (32 Bits) (Optional)

The purpose of this field is to provide a standardized mechanism for reporting a small number of real-time functions (e.g., telecommand verification or spacecraft clock calibration). The leading bit of the field (Bit 0) is a “Type” flag that indicates which function is being reported.

A “Type 1” report (Bit 0 = "0") contains a “Command Link Control Word” which is used to provide acceptance reporting for spacecraft that are compatible with the Transfer Frame layer of the Packet Telecommand concept: the internal format of the Command Link Control Word is fully defined in Reference [3]. The format of a “Type 2” report (Bit 0 = "1") is currently undefined and is reserved for future application.

The presence or absence of this field is signalled by the Operational Control Field Flag within the Transfer Frame Primary Header. If present, the field must occur within EVERY frame transmitted through a physical data channel.

If the optional Frame Error Control Word is NOT present, the Operational Control Field occupies the four trailing octets of the Transfer Frame: if the Error Control Word IS present, the field is displaced towards the beginning of the frame by two octets.

5.5.2 FRAME ERROR CONTROL WORD (16 Bits) (Optional)

The purpose of this **16-bit field** is to provide a capability for detecting errors which may have been introduced into the frame during the data handling processes.

If the Transfer Frame is NOT Reed-Solomon encoded, the presence of the Frame Error Control Word is mandatory. The field is optional if the Frame is synchronously contained within the data space of a Reed-Solomon codeblock. If present, the field occupies the two trailing octets of the Transfer Frame. Presence or absence of the field is implied from the Spacecraft Identifier and is specified to the ground network as a mission set-up parameter.

The cyclic redundancy code is characterized as follows:

(1) The generator polynomial is:

$$g(x) = x^{16} + x^{12} + x^5 + 1$$

(2) Both encoder and decoder are initialized to the “all-ones” state for each transfer frame.

(3) Parity (P) generation is performed over the data space "D" as shown for each of the two options in Figure 5-3. Note that option A is preferred for new designs.

Option A: "D" covers the transfer frame (less the final 16-bit parity field) and EXCLUDES the attached sync marker.

Option B: "D" covers the transfer frame (less the final 16-bit parity field) and INCLUDES the attached sync marker.

(4) The generated parity symbols are then inserted into the parity check field which occupies the final 16 bits of the transfer frame.

The procedure for generating the parity symbols is described in Reference [4].

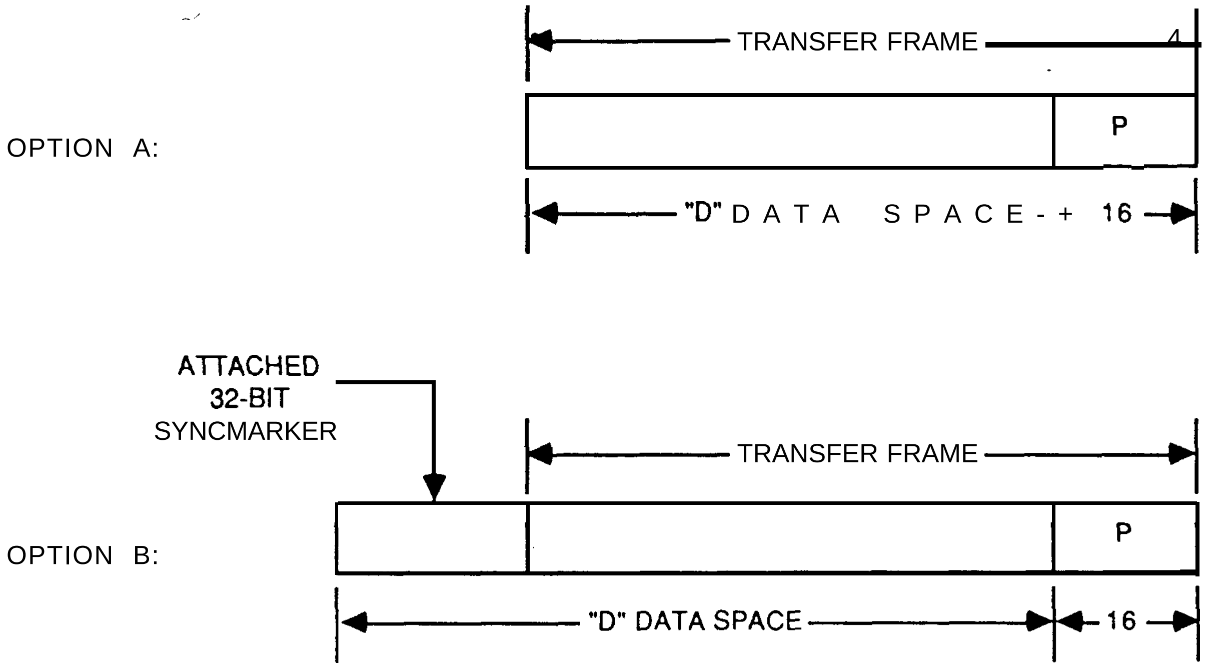


Figure 5-3: Fields Over Which Parity Is Generated